Project Report
For
Line Encoder

Sahil Gupta - 2017BITE045
Kanav Samyal - 2017BITE020

Data Communication
Department of Information Technology
National Institue of Technology Srinagar

20 Dec 2019

Abstract

A computer network is designed to send information from one point to another. This information needs to be converted to either a digital signal or an analog signal for transmission. In this project we have shown varoius ways in which a digital data can be converted to a digital signal.

Line coding is used to convert *digital data* to a *digital signal*. Several common schemes are NRZ, AMI, B8ZS, Manchester, etc.

This project has been done using **Python** and its library **MATPLOTLIB** on Jupyter Notebook Environment.

Contents

1 Signal Encoding **2**
- 1.1 Line Encoding . . . 2
  - 1.1.1 What is it? . . . 2
  - 1.1.2 Line Coding Schemes . . . 2
- 1.2 Scrambling . . . 3
  - 1.2.1 What is it? . . . 3
  - 1.2.2 Scrambling Schemes . . . 3
- 1.3 Definition of Digital Signal Encoding Formats . . . 3

2 Line Encoder : How to use the code **5**

Chapter 1

Signal Encoding

1.1 Line Encoding

1.1.1 What is it?

Line coding is the process of converting digital data to digital signals. We assume that data, in the form of text, numbers, graphical images, audio, or video, are stored in computer memory as sequences of bits. Line coding converts a sequence of bits to a digital signal. At the sender, digital data are encoded into a digital signal; at the receiver, the digital data are recreated by decoding the digital signal.

1.1.2 Line Coding Schemes

1. ***Unipolar***
    - *NRZ*
2. ***Polar***
    - *NRZ*
    - *RZ*
    - *Biphase*
        - Manchester
        - Differential Manchester
3. ***Bipolar***
    - *AMI*
    - *Pseudoternary*

1.2 Scrambling

1.2.1 What is it?

We are looking for a solution that substitutes long zero-level pulses with a combination of other levels to provide synchronization. One solution is called **scrambling**.

1.2.2 Scrambling Schemes

1. B8ZS
2. HDB3

1.3 Definition of Digital Signal Encoding Formats

1. Nonreturn to Zero-Level (NRZ-L)
   - 0 = high level
   - 1 = low level
2. Nonreturn to Zero Inverted (NRZI)
   - 0 = no transition at beginning of interval (one bit time)
   - 1 = transition at beginning of interval
3. Bipolar-AMI
   - 0 = no line signal
   - 1 = positive or negative level, alternating for successive ones
4. Pseudoternary
   - 0 = positive or negative level, alternating for successive zeros
   - 1 = no line signal
5. Manchester
   - 0 = transition from high to low in middle of interval
   - 1 = transition from low to high in middle of interval

6. Differential Manchester
    - Always a transition in middle of interval
    - 0 = transition at beginning of interval
    - 1 = no transition at beginning of interval
7. B8ZS
    - Same as bipolar AMI, except that any string of eight zeros is replaced by a string with two code violations
8. HDB3
    - Same as bipolar AMI, except that any string of four zeros is replaced by a string with one code violation

Chapter 2

Line Encoder : How to use the code

The whole code is written in Python language and can be run using Jupyter Notebook.
Matplotlib is used to make the graph that will show the digital signal produced for the given digital data stream in the encoding format.

Given steps can be followed :

1. Launch Jupyter Notebook.
2. Open the file in it.
3. Wait for it to load the libraries.
4. Run the code.
5. Enter the number of the Encoding Scheme you want to perform.
6. Enter the data stream *'without spaces'* in the dialog box.
7. The graph formed will show the digital data in the digital signal in the desired encoding format.

```
Please enter the encoding you want to perform:
1 - NRZL
2 - NRZI
3 - Manchester
4 - Differential Manchester
5 - Bipolar-AMI
6 - B8ZS
7 - HDB3
0 - Quit
Enter your choice : 3
Enter data stream : 10110111010001
```

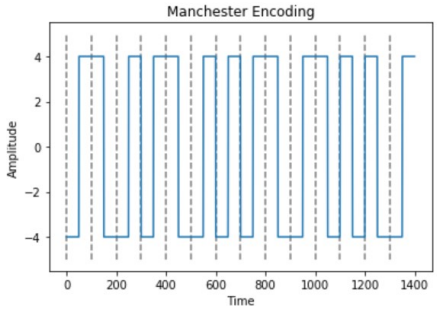

Figure 2.1: An example of Manchester Encoding

```
Please enter the encoding you want to perform:
1 - NRZL
2 - NRZI
3 - Manchester
4 - Differential Manchester
5 - Bipolar-AMI
6 - B8ZS
7 - HDB3
0 - Quit
Enter your choice : 5
Enter data stream : 110100111
```

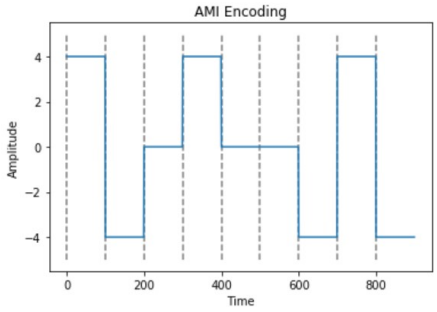

Figure 2.2: An example of Bipolar AMI

```
Please enter the encoding you want to perform:
1 - NRZL
2 - NRZI
3 - Manchester
4 - Differential Manchester
5 - Bipolar-AMI
6 - B8ZS
7 - HDB3
0 - Quit
Enter your choice : 7
Enter data stream : 11100001001100000000111
```

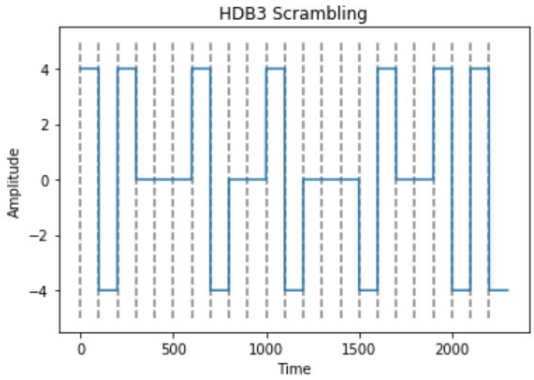

Figure 2.3: An example of HDB3